12.1. Настройка и компиляция проекта в Eclipse, использующего STemWin Library, для демо-платы STM3241G_EVAL.

1) Создать новый проект с именем stemwin. File -> New -> C Project.

🚯 C Project				
C Project Create C project of selected type				
Project name: stemwin Use default location Location: D:\Konstantinov\android_work	(space\stemwin			Browse
Project type:	To	olchains:		
ARM Cross Target Application Empty Project Hello ARM World C Project ARM Cross Target Static Library ARM Cross Target Static Library ARM Cross Target Static Library		ARM Windows ARM Windows ARM Windows ARM Windows ARM Windows ARM Windows ARM Windows ARM Windows ARM Windows	GCC (GNUARM, GCC (Linaro AAro GCC (Linaro AAro GCC (Linaro GNU GCC (Sourcery Li GCC (Sourcery Li GCC (Sourcery Li GCC (Sourcery Li GCC (Sourcery Li GCC (Yagarto) GCC (devkitPro)	WinARM) ch64 Bare) ch64 Linux) JEABIHF) ite Bare) ite Linux) imbedded)
Show project types and toolchains onl	y if they are sup	ported on the pla	ltform	
?	< Back	Next>	Finish	Cancel

- 2) Нажать Finish.
- 3) Пока проект выглядит так.
- 🖻 😂 stemwin
- 🗄 👘 Includes
- В «Includes» содержаться пути к включаемым каталогам GNU Tools ARM Embedded.

4) Создать в проекте каталог CMSIS.

Поместить в каталог CMSIS 9 файлов:

core	_cm4_simd.h
core	_cm4.h
core	_cmFunc.h
core	_cmInstr.h

startup_stm32f4xx.S
stm32f4xx_conf.h
stm32f4xx.h
system_stm32f4xx.c
system_stm32f4xx.h

Расширение .S обязательно с большой буквы! Иначе компилятор не поймет.

stm32f4xx_conf.h на самом деле не принадлежит CMSIS-CORE, но д. б. в этом каталоге т. к. на него ссылается stm32f4xx.h, если определена USE_STD_PERIPH_DRIVER.

B stm32f4xx_conf.h — включения всех стандартных заголовочных файлов периферии.

Все файлы м. взять, например, из STM32F4xx_DSP_StdPeriph_Lib_V1.1.0. Свободно загружается с официального сайта st.com

5) Создать в проекте каталог StdPeripheralDriver.

Поместить во вновь созданный каталог StdPeripheralDriver каталоги inc и src из каталога STM32F4xx_StdPeriph_Driver – стандартные драйверы периферии микроконтроллера. М. взять, например, из STM32F4xx_DSP_StdPeriph_Lib_V1.1.0. Свободно загружается с официального сайта st.com.

inc	заголовочные файлы драйверов
src	исходные файлы драйверов

```
6) Добавить к проекту новый C-файл main.c
Содержание файла main.c минималистичное:
int main(void)
{
    return 0;
}
Функция main – точка входа в программу.
```

7) Добавить к проекту файл линкера stm32 flash.ld

После этого шага проект д. выглядеть так:

🖻 📂 ster	nwin
÷ 👘	Includes
÷	CMSIS
÷	StdPeripheralDriver
÷c	main.c
🧭	stm32_flash.ld

8) Настроить проект Project->Properties Выбрать процессор Cortex-M4



Добавить «определенные» символы компилятора STM32F4XX и USE_STDPERIPH_DRIVER



Добавить включаемые каталоги компилятора: CMSIS и StdPeripheralDriver/inc

Operation of the stemwin in the stem win the stem win in the s		
type filter text	Settings	$\leftarrow \bullet \bullet \bullet \bullet \bullet \bullet$
 Resource Builders C/C++ Build Build Variables 	Configuration: Debug [Active]	Manage Configurations
 → Diala Validations → Discovery Options → Environment → Logging → Settings → Tool Chain Editor ⊕ C/C++ General → Project References → Refactoring History → Run/Debug Settings → Task Tags ⊕ Validation 	Tool Settings Build Steps Build Artifact Binary Parsers Target Processor Debugging Build Artifact Binary Parsers Additional Tools ARM Windows GCC Assembler (Tools For Embedded) Directories Directories Binary Parsers Binary Parsers Warnings Miscellaneous Binary Parsers Directories Binary Parsers Directories Binary Parsers Binary Parsers Binary Parsers Warnings Miscellaneous Binecellaneous Binecellaneous Binecellaneous Binecellaneous Miscellaneous Binecellaneous Miscellaneous Binecellaneous Miscellaneous Binecellaneous Miscellaneous Binecellaneous Miscellaneous Binecellaneous Binecellaneous Binecellaneous	
		Bestore Defaults Apply
?		OK Cancel

Указать скрипт линкера stm32_flash.ld. Не использовать стандартные стартовые файлы (у нас свой - startup stm32f4xx.S)



9) Выполнить компиляцию проекта

Project -> Build Project

🔝 Problems 🖉 Tasks 📃 Console 🕱 🔲 Properties 🕸 Debug 🚟 Disassembly 🕮 EmbSys Registers 📱 Devices 🛃 Projects 式 Progress	- 0		
🕹 🗘 😓 🖬 🔝 🖦 🖃 🖛	1		
CDT Build Console [stemwin]			
Building file:/StdPeripheralDriver/src/stm32f4xx_gpio.c			
Invoking: ARM Windows GCC C Compiler (Tools For Embedded)			
arm-none-eabi-gcc -DSTM32F4XX -DUSE STDPERIPH DRIVER -I"D:\Konstantinov\android workspace\stemwin\CMSIS"			
-I"D:\Konstantinov\android workspace\stemwin\StdPeripheralDriver\inc" -00 -ffunction-sections -fdata-sections -Wall			
-Wa,-adhlns="StdPeripheralDriver/src/stm32f4xx gpio.o.lst" -c -fmessage-length=0 -MMD -MP			
-MF"StdPeripheralDriver/src/stm32f4xx gpio.d" -MT"StdPeripheralDriver/src/stm32f4xx gpio.d" -mcpu=cortex-m4 -mthumb			
g3 -o "StdPeripheralDriver/src/stm32f4xx gpio.o" "/StdPeripheralDriver/src/stm32f4xx gpio.c"			
	•		

В процессе компиляции было показано предупреждение о не используемой переменной tmp_size. Не обращаем на это предупреждение внимание.

Building file: ../StdPeripheralDriver/src/stm32f4xx flash.c

Invoking: ARM Windows GCC C Compiler (Tools For Embedded)

arm-none-eabi-gcc -DSTM32F4XX -DUSE_STDPERIPH_DRIVER -I"D:\Konstantinov\android_workspace\stemwin\CMSIS"

ann-hohe-cableget = 25H32F4XA = DSE_STM2F4XFH_DXAVFA = I biological file of the section of the s

- ../StdPeripheralDriver/src/stm32f4xx_flash.c: In function 'FLASH_EraseAllSectors':

../StdPeripheralDriver/src/stm32f4xx_flash.c:434:12: warning: variable 'tmp_psize' set but not used

Finished building: ../StdPeripheralDriver/src/stm32f4xx_flash.c

🔝 Problems 🖉 Tasks 📮 Console 🕴 🗔 Properties 🎋 Debug 🚟 Disassembly 🕮 EmbSys Registers 📮 Devices 🔂 Projects 🕞 Projects 🕞 Progress
CDT Build Console (stemwin)
Building target: stemwin.elf
Invoking: ARM Windows GCC C Linker (Tools For Embedded)
arm-none-eabi-gcc -T"D:\Konstantinov\android workspace\stemwin\stm32 flash.ld" -nostartfiles -Xlinkergc-sections
-W1,-Map,"stemwin.map" -mcpu=cortex-m4 -mthumb -g3 -o "stemwin.elf" ./StdPeripheralDriver/src/misc.o
./StdPeripheralDriver/src/stm32f4xx adc.o ./StdPeripheralDriver/src/stm32f4xx can.o
./StdPeripheralDriver/src/stm32f4xx crc.o ./StdPeripheralDriver/src/stm32f4xx cryp.o
./StdPeripheralDriver/src/stm32f4xx cryp aes.o ./StdPeripheralDriver/src/stm32f4xx cryp des.o
./StdPeripheralDriver/src/stm32f4xx cryp tdes.o ./StdPeripheralDriver/src/stm32f4xx dac.o
./StdPeripheralDriver/src/stm32f4xx_dbgmcu.o/StdPeripheralDriver/src/stm32f4xx_dcmi.o
./StdPeripheralDriver/src/stm32f4xx_dma.o ./StdPeripheralDriver/src/stm32f4xx_exti.o
./StdPeripheralDriver/src/stm32f4xx_flash.o/StdPeripheralDriver/src/stm32f4xx_fsmc.o
./StdPeripheralDriver/src/stm32f4xx_gpio.o ./StdPeripheralDriver/src/stm32f4xx_hash.o
./StdPeripheralDriver/src/stm32f4xx_hash_md5.o ./StdPeripheralDriver/src/stm32f4xx_hash_sha1.o
./StdPeripheralDriver/src/stm32f4xx_i2c.o ./StdPeripheralDriver/src/stm32f4xx_iwdg.o
./StdPeripheralDriver/src/stm32f4xx_pwr.o ./StdPeripheralDriver/src/stm32f4xx_rcc.o
./StdPeripheralDriver/src/stm32f4xx_rng.o ./StdPeripheralDriver/src/stm32f4xx_rtc.o
./StdPeripheralDriver/src/stm32f4xx_sdio.o ./StdPeripheralDriver/src/stm32f4xx_spi.o
./StdPeripheralDriver/src/stm32f4xx_syscfg.o ./StdPeripheralDriver/src/stm32f4xx_tim.o
./StdPeripheralDriver/src/stm32f4xx_usart.o ./StdPeripheralDriver/src/stm32f4xx_wwdg.o ./CMSIS/startup_stm32f4xx.o
./CMSIS/system_stm32f4xx.o ./main.o
Finished building target: stemwin.elf
Invoking: ARM Windows GNU Create Flash Image (Tools For Embedded)
arm-none-eabi-objcopy -O ihex "stemwin.elf" "stemwin.hex"
Finished building: stemwin.hex
Invoking: ARM Windows GNU Create Listing (Tools For Embedded)
arm-none-eabi-objdump -h -S "stemwin.elf" > "stemwin.lst"
Finished building: stemwin.lst
Invoking: ARM Windows GNU Print Size (Tools For Embedded)
arm-none-eabi-sizeformat=berkeley "stemwin.elf"
text data bss dec hex filename
1048 0 1100 2148 864 stemwin.elf
Finished building: stemwin.siz
15:33:27 Build Finished (took 1m:38s.937ms)
×

Компиляция прошла успешно, создан исполняемый файл stemwin.elf.

^{[-}Wunused-but-set-variable]

10) Структура проекта

÷	🞏 stemwin
	🖻 🐝 Binaries
1	🗄 🕸 stemwin.elf - [arm/le]
	🗄 👘 Includes
	🗄 🔁 CMSIS
ŀ	🗄 🗁 Debug
	🗄 🚌 StdPeripheralDriver
	🗄 🔂 main.c
	🦾 🥑 stm32_flash.ld

Далее включим в проект библиотеку STemWin.

11) Создать в проекте каталог GUI, куда будем складывать все, относящееся к Graphical User Interface.

12) Поместить в каталог GUI библиотеку STemWin522_CM4_GCC. а из каталога STemWin_Library_V1.1.0\Libraries\STemWinLibrary522\Lib\

Добавить префикс lib к имени библиотеки: таким образом переименовать библиотеку из STemWin522_CM4_GCC.a в libSTemWin522_CM4_GCC.a

Это нужно для особенностей компилятора GCC - у библиотеки д. б. префикс lib.

Добавить включение библиотеки libSTemWin522_CM4_GCC.а и путь к ней в путях поиска библиотек линкера.

Operation of the standard stand Standard standard stand Standard standard stand Standard standard stand Standard standard stand Standard standard standard standard standard standard standard standar		
type filter text	Settings	$\leftarrow \tau \Rightarrow \tau =$
Resource Builders C/C++ Build Build Variables Discovery Options Environment Logging Settings	Configuration: Debug [Active]	Manage Configurations
Tool Chain Editor → Tool Chain Editor → Project References → Refactoring History → Run/Debug Settings → Task Tags → Validation	Target Processor Debugging Additional Tools Additional Tools Preprocessor Warnings Warnings Warnings Directories Warnings Directories Warnings Directories Warnings Biscellaneous Beneral Warnings Bout Duput Section Windows GNU Create Listing (Tools For Embedded) Beneral Warnings ARM Windows GNU Print Size (Tools For Embedded) Beneral	Library search path (L) * (workspace_loc:/\${Pro[Name)/GUI)''
		Restore Defaults Apply
?		OK Cancel

13) Поместить в каталог GUI каталог

STemWin_Library_V1.1.0\Libraries\STemWinLibrary522\inc, содержащий заголовочные файлы библиотеки STemWin.





14) Создать каталог STM32_EVAL в проекте

В этом каталоге будет все, что относиться к самой отладочной плате STM3241G_EVAL.

15) Добавить в каталог STM32_EVAL каталог

STemWin_Library_V1.1.0\Utilities\STM32_EVAL\STM3240_41_G_EVAL, содержащий файлы настройки периферии для отладочной платы STM3241G_EVAL. Хотя некоторые файлы из него и не нужны, но поскольку он стандартный, оставим все как есть.

Добавить в пути компилятора путь к каталогу STM3241G_EVAL.

16) Добавить в каталог STM32_EVAL каталог STemWin_Library_V1.1.0\Utilities\STM32_EVAL\Common, содержащий общие настройки для отладочных плат STM32_EVAL.

В каталоге Common нужно переименовать файл lcd_log_conf_template.h в
lcd_log_conf.h и в строке

```
#include "stm32xxx_eval_lcd.h" /* replace 'stm32xxx' with your
EVAL board name, <u>ex</u>:
stm3210c_eval_lcd.h */
Заменить stm32xxx eval lcd.h на stm324xg eval lcd.h
```

Добавить в пути компилятора путь к каталогу Common.

17) Скопировать в каталог GUI каталог STemWin_Library_V1.1.0\Project\STM324xG-EVAL\Standalone\Config, содержащий файлы

GUIConf_stm3240g_eval.h/.c LCDConf_stm3240g_eval.h/.c global_includes.h

переименовать файлы

GUIConf_stm3240g_eval.h/.c BGUIConf.h/.c
LCDConf_stm3240g_eval.h/.c BLCDConf.h/.c

Добавить в пути компилятора путь к каталогу Config.

18) Скопировать в каталог проекта каталог STemWin_Library_V1.1.0\Project\STM324xG-EVAL\Standalone\Demo, содержащий исполняемые файлы демонстрационных экранов.

Добавить в пути компилятора путь к каталогу Demo.

19) Скопировать из каталога STemWin_Library_V1.1.0\Project\STM324xG-EVAL\Standalone\User следующие файлы:

в каталог Demo:

bsp.h bsp.c

в каталог проекта:

main.c
stm32xxx_it.c
stm32xxx_it.h

Будет предложено заменить main.c. Согласиться.

20) Начать компиляцию проекта.



Компилятор сообщит об ошибке libSTemWin522_CM4_GCC.a uses VFP register arguments, stemwin.elf does not. Что говорит о том, что библиотека STemWin использует регистры FPU (Floating Point Unit), так как при компиляции библиотеки разработчиками была указана поддержка FPU. Но в нашем проекте мы не указали использовать FPU. Сделаем это сейчас.

Нужно указать использование FPU. Float ABI: FP instructions. FPU Type: FPv4 SP D16. Именно эти параметры указаны, так как STM32F417 имеет FPU и этот FPU реализует FPv4-SP расширение.



21) Компилируем еще раз и снова ошибка



Oшибка multiple definition of "ASCII16x24_Table" и т. д. из-за двойного включения файла font.c:

включения файла топс. С:

I) fonts.c компилируется из каталога STM32_EVAL/Common

II) fonts.c включается в файле stm324xg_eval_lcd.c

Следовательно, нужно исключить файл из компиляции, но чтобы его содержимое было в файле stm324xg eval lcd.c.

Для этого переименовываем fonts.cвfonts_c.hиделаем такое же переименование внутри файла stm32xg_eval_lcd.c #include "../Common/fonts.c" -> #include "../Common/fonts c.h"

22) Компилируем еще раз и снова ошибки



is already declared in stm32_eval_audio_codec.h) */

void EVAL_AUDIO_TransferComplete_CallBack(uint32 t pBuffer,

```
uint32_t Size)
{}
```

Поскольку функция связана с аудио, а мы подключаем графическую библиотеку, то не особо заботимся о ее содержимом.

```
II) undefined reference to EVAL sEE_TIMEOUT_UserCallback()-
неопределенная ссылка на функцию sEE_TIMEOUT_UserCallback(). Открываем файл
stm324xg_eval_i2c_ee.h и раскомментируем
#define USE_DEFAULT_TIMEOUT_CALLBACK, как и написано в самом файле:
    /* <u>Uncomment</u> the following line to use the default
    sEE_TIMEOUT_UserCallback() function implemented in
    stm32_evel_i2c_ee.c file. sEE_TIMEOUT_UserCallback() function
    is called whenever a timeout condition <u>occure</u> during
    communication (waiting on an event that doesn't occur, bus
    errors, busy devices ...). */
/* #define USE_DEFAULT_TIMEOUT_CALLBACK */
```

III) undefined reference to OS_TimeMS - неопределенная ссылка на OS_TimeMS. В самом файле stm32xxx_it.c эта переменная объявлена как: extern __IO int32_t OS_TimeMS;

Истинное определение в файле GUI_X.с, которого в нашем проекте пока нет. Скопировать из каталога STemWin_Library_V1.1.0\Libraries\STemWinLibrary522\OS файл GUI_X.с в каталог GUI проекта.

23) Компилируем еще раз и снова ошибки

```
🔐 Problems 🖉 Tasks 📮 Console 🛿 🔲 Properties 🎋 Debug 🚟 Disassembly 🚥 EmbSys Registers  🖶 Devices 🔁 Projects 🖏 Projects
                                                                                                                      - -
                                                                                           🕂 🗘 🔄 🔜 🗛 🛃 🗉 - 📬 -
CDT Build Console [stemwin]
d:/stm/gnu tools arm embedded/4.7
                                                                                                                         .
2013q3/bin/../lib/gcc/arm-none-eabi/4.7.4/../../arm-none-eabi/lib/armv7e-m/fpu\libg.a(lib_a-sbrkr.o): In
function `_sbrk_r':
sbrkr.c:(.text._sbrk_r+0xc): undefined reference to `_sbrk'
d:/stm/gnu tools arm embedded/4.7
2013q3/bin/../lib/gcc/arm-none-eabi/4.7.4/../../arm-none-eabi/lib/armv7e-m/fpu/libm.a(lib a-w sqrt.o): In
function `sqrt':
w_sqrt.c:(.text.sqrt+0x9c): undefined reference to `__errno'
                                                     __errno'
w_sqrt.c:(.text.sqrt+0xa6): undefined reference to
collect2.exe: error: ld returned 1 exit status
make: *** [stemwin.elf] Error 1
09:28:11 Build Finished (took 5s.594ms)
```

Здесь начинается самое интересное.

I) undefined reference to '_sbrk' - неопределенная ссылка на _sbrk.

_sbrk – функция выделения памяти в куче (heap).

```
Напишем свою функцию sbrk в файле main.c:
extern int HEAP_START;
caddr t sbrk ( int incr )
{
  static unsigned char *heap = 0;
  unsigned char *prev heap;
  if (heap == 0) {
    heap = (unsigned char *) & HEAP START;
  }
  prev heap = heap;
  /* check removed to show basic approach */
  heap += incr;
  return (caddr t) prev heap;
}
и добавим ____HEAP_START в файл линкера в секцию .__user_heap_stack:
  /* User heap stack section, used to check that there is
enough RAM left */
  ._user_heap_stack :
  {
    = ALIGN(4);
```

```
PROVIDE (___HEAP_START = . );  /* !!! */
. = . + __Min_Heap_Size;
. = . + __Min_Stack_Size;
. = ALIGN(4);
```

} >RAM

```
II) undefined reference to __errno - неопределенная ссылка на __errno.
Напишем функцию-заглушку в файле main.c
void __errno(void) //dummy
{
}
```

24) На этом все. Компилируем и загружаем stemwin.elf в демо-плату и вуаля.

🚯 Problems 🖉 Tasks 🖳 Console 🕺 🔲 Properties 🎋 Debug 🎫 Disassembly 🚥 EmbSys Registers 🔒 Devices 🔂 Projects 🖏 Progress	7 🗖
🕂 🗘 🖓 🖓 🖬 🖬 🖬 🖓 🗢	1
CDT Build Console [stemwin]	
Building target: stemwin.elf	
Invoking: ARM Windows GCC C Linker (Tools For Embedded)	
arm-none-eabi-gcc -T"D:\Konstantinov\android_workspace\stemwin\stm32_flash.ld" -nostartfiles -Xlinkergc-sections	
-L"D:\Konstantinov\android_workspace\stemwin\GUI" -W1,-Map,"stemwin.map" -mcpu=cortex-m4 -mthumb -mfloat-abi=hard	
-mfpu=fpv4-sp-d16 -g3 -o "stemwin.elf" ./StdPeripheralDriver/src/misc.o ./StdPeripheralDriver/src/stm32f4xx_adc.o	
./StdPeripheralDriver/src/stm32f4xx_can.o ./StdPeripheralDriver/src/stm32f4xx_crc.o	
./StdPeripheralDriver/src/stm32f4xx_cryp.o ./StdPeripheralDriver/src/stm32f4xx_cryp_aes.o	
./StdPeripheralDriver/src/stm32f4xx_cryp_des.o ./StdPeripheralDriver/src/stm32f4xx_cryp_tdes.o	
./StdPeripheralDriver/src/stm32f4xx_dac.o ./StdPeripheralDriver/src/stm32f4xx_dbgmcu.o	
./StdPeripheralDriver/src/stm32f4xx_dcmi.o ./StdPeripheralDriver/src/stm32f4xx_dma.o	- 11
./StdPeripheralDriver/src/stm32f4xx_exti.o ./StdPeripheralDriver/src/stm32f4xx_flash.o	- 11
./StdPeripheralDriver/src/stm32f4xx_fsmc.o ./StdPeripheralDriver/src/stm32f4xx_gpio.o	- 11
./StdPeripheralDriver/src/stm32f4xx_hash.o ./StdPeripheralDriver/src/stm32f4xx_hash_md5.o	
./StdPeripheralDriver/src/stm32f4xx_hash_sha1.o ./StdPeripheralDriver/src/stm32f4xx_i2c.o	
./StdPeripheralDriver/src/stm32f4xx_iwdg.o ./StdPeripheralDriver/src/stm32f4xx_pwr.o	
./stdPeripheralDriver/src/stm32f4xx_rcc.o ./stdPeripheralDriver/src/stm32f4xx_rng.o	
./stdPeripheralDriver/src/stm32f4xx_rtc.o ./stdPeripheralDriver/src/stm32f4xx_sdio.o	- 11
./stdPeripheralDriver/src/stm32f4xx_spi.o ./stdPeripheralDriver/src/stm32f4xx_syscig.o	- 11
/StdPeripheralDriver/src/stm30f4xx tim.o./StdPeripheralDriver/src/stm32f4xx usatt.o	- 11
/StareripheralDriver/src/stm32f4xx wwag.o ./Sim32 Eval/Sim3240 41 G Eval/Stm324xg eval.o	- 11
(STN92 EVAL/SIN3240 41 G EVAL/SIN324Xg EVAL audio Codec. D. (SIN32 EVAL/SIN3240 41 G EVAL/SIN324Xg EVAL ISMC SIAM.O	- 11
/SINS2_EVEL/SINS240_41_G_EVEL/SUNS24Xg_EVEL_122_EVEL_SINS240_41_G_EVEL/SUNS24Xg_EVEL_DEUD	
/SINS2_DVAL/SINS24V0_11_G_VVAL/SUNS24X9_EVAL_ICA.0 / SINS2_DVAL/SINS24V0_11_G_VVAL/SUNS24X9_EVAL_SUNS	
//new_/CULDEWO hotiolicedTayt o //bwo/CULDEWO hytomative o //bwo/CULDEWO Referable o //bwo/CULDEWO bitwoot o	
/ Demo/GUIDENO_ALCIARIASECTEX	
/ Demo/GUIDEMO_ICONVIEW of / Demo/GUIDEMO_INGREE or of / Demo/GUIDEMO_INGROUP of AUTOMOUSTIC / Demo/GUIDEMO_ISTATION / DEMO/GUIDEMO_INGROUP of AUTOMOUSTIC / DEMO/GUIDEMO	.0
/ Demo/GUIDEMO_RedialMenu.g. / Demo/GUIDEMO_REGUETE.g. / Demo/GUIDEMO_Skinning.g. / Demo/GUIDEMO_Sheed g	
./Demo/GHIDEMO_Meeter.o./Demo/GHIDEMO_Start.o./Demo/GHIDEMO_TransmarentDialog.o./Demo/GHIDEMO_Treeview.o.	
./Demo/GUIDEMO_VScreen.o./Demo/GUIDEMO_WashingMachine.o./Demo/GUIDEMO_ZonMARDATE.o./Demo/JScreen.o.	
./CMSTS/startup_stm32f4xx.o./CMSTS/system_stm32f4xx.o./main.o./stm32xxx it.oISTembin522 CM4 GCC	
Finished building target: stemwin.elf	
	•

И еще, у меня получилось так, что символ определен, программа компилируется, а Eclipse подчеркивает его как symbol could not be resolved. Решение данной проблемы – проверьте, чтобы в настройках проекта в C/C++ general -> Indexer стояли все галочки.

Окончательная структура проекта:

🖻 🞏 stemwin 🗄 🐇 Binaries 🗄 👘 Includes 🗄 🗁 CMSIS 🗄 🗁 Debug 🗄 🗁 Demo 🚊 🕞 GUL 🗄 🗁 Config 🗄 🗁 inc 🗄 🔂 GUI_X.c 🔚 🔚 libSTemWin522_CM4_GCC.a 🗄 🗁 StdPeripheralDriver 🚊 🎓 STM32_EVAL 🗄 🗁 Common 🗄 🔁 STM3240_41_G_EVAL 🗄 🔂 main.c 🗄 🖸 stm32xxx_it.c 🗄 🖬 stm32xxx_it.h 🦾 🖌 stm32_flash.ld

Содержание каталога Demo:

- 🖻 🗁 Demo 🗄 🛛 💽 bsp.c. 🗄 🖬 庙 bsp.h 🗄 🔂 GUIDEMO_AntialiasedText.c 🗄 🔂 GUIDEMO_Automotive.c 😟 🗟 GUIDEMO_BarGraph.c 🗄 🔂 GUIDEMO_Bitmap.c 🗄 🔂 GUIDEMO_ColorBar.c 🗄 🔂 GUIDEMO_Conf.c 🗄 🔂 GUIDEMO_Cursor.c 🗄 🔂 GUIDEMO_Fading.c 🖻 🔂 GUIDEMO_Graph.c 😟 🔂 GUIDEMO_IconView.c 🗄 🔂 GUIDEMO_ImageFlow.c 🗄 🗟 GUIDEMO_Intro.c 🗄 🔂 GUIDEMO_Listview.c 🗄 🔂 GUIDEMO_RadialMenu.c 🗄 🗟 GUIDEMO_Resource.c 🗄 🔂 GUIDEMO_Skinning.c E-GUIDEMO Speed.c 🗄 🔂 GUIDEMO_Speedometer.c 🗄 🔂 GUIDEMO_Start.c 🗄 🔂 GUIDEMO_TransparentDialog.c 🗄 🖻 GUIDEMO_Treeview.c 🗄 🔂 GUIDEMO_VScreen.c 🗄 🔂 GUIDEMO_WashingMachine.c 🗄 🔂 GUIDEMO_ZoomAndRotate.c 🗄 🔂 GUIDEMO.c.
 - 🗄 🔚 GUIDEMO.h

Содержание каталога GUI:



Содержание каталога STM32_EVAL:

🗄 😥 STM32_EVAL 🖻 🧀 Common 🗄 🔚 fonts_c.h 🗄 庙 fonts.h 🗄 🔚 Icd_log_conf.h . 🗄 🔂 Icd_log.c 🗄 🔚 Icd_log.h 🗄 🔚 stm32_eval_legacy.h 🖻 🎓 STM3240_41_G_EVAL 🗄 🖻 stm324xg_eval_audio_codec.c 🗄 🔚 stm324xg_eval_audio_codec.h 🗄 🔂 stm324xg_eval_fsmc_sram.c 🗄 🖬 stm324xg_eval_fsmc_sram.h 🗄 🚾 stm324xg_eval_i2c_ee.c 连 📠 stm324xg_eval_i2c_ee.h 🗄 🔂 stm324xg_eval_ioe.c 🗈 🗄 🔚 stm324xg_eval_ioe.h 🗄 🔂 stm324xg_eval_lcd.c 🗈 🗄 🛅 stm324xg_eval_lcd.h 🗄 🖻 stm324xg_eval_sdio_sd.c 🛛 😟 📠 stm324xg_eval_sdio_sd.h 🗄 🖸 stm324xg_eval.c 🗄 🔚 stm324xg_eval.h 🔤 🎯 Release_Notes.html

Содержание каталога CMSIS:

🖻 🗁 CMSIS

🗄 🛅 core_cm4_simd.h

- 🗄 🖬 core_cm4.h
- 🗄 🔚 core_cmFunc.h
- 🗄 🛅 core_cmInstr.h
- 🗄 🔝 startup_stm32f4xx.S
- 🗄 🖬 stm32f4xx_conf.h.
- 🗄 🔚 stm32f4xx.h
- 🗄 🖻 system_stm32f4xx.c
- 🗄 🖬 system_stm32f4xx.h 🗄

Фотографии отладочной платы STM3241G_EVAL, на которой запущено демонстрационное приложение, использующее библиотеку STemWin.

